Day 13 Task

Extending previous assignment. Your application should mimic an ATM machine while doing any transaction. Make use of appropriate collections (list, set or map) for a Passbook (Account summary). Refer your bank passbook and add all the required parameters (or variables) available on it, especially the date-and-time. Finally print the passbook (sorted by date&time).

Don’t get confused between mini statement and Account summary. Mini statement should provide information of only last 5 transactions based on date-and-time.

Make your application as user friendly as possible.

Prepare a word document, explain below:

- Explain your understanding of given assignment description; consider previous assignment descriptions also. (we call it “Functional specification” document)

- Explain your logic of implementation (we call it “technical specification” document)

- Which collection you have used? And why?

- What improvements has been done since your last submission?

**Functional Specifications:**

In this task we have to prepare software which can perform operations:

1. Credit – Deposit Money

2. Debit – Withdraw Money

3. Generate Passbook – A passbook stores information of all the transactions performed by the user (like particulars, date, final balance of each transaction).

4. Generate Mini-Statement – A mini statement contains information of last 5 transactions.

**Technical Specifications & Collection used:**

**1. Queue (Mini Statement):** It is suitable to use because of its FIFO nature. On applying the remove() method it will delete the transaction at 0 index (i.e. the oldest transaction) and a new transaction will be inserted at the end.

Queue<String> miniStatement = new LinkedList<>();

**2. HashMap (Passbook):** A Passbook works (key, value) structure.

Key stores the serial no of transactions. And value are transactions itself which stores the information of all operations performed.

HashMap<Integer, Transactions> passbook = new HashMap<>();

**Here Transactions is a class.** Every object of transactions will have information of each operation (credit, debit) performed.

1. Particulars

2. Cheque number

3. Debit amount

4. Credit amount

5. Final Balance

6. Date

7. Time

Example: if user credits Rs. 100 into the account. Then the passbook will look like

Passbook<1, new Transactions(“cash”, - , 0, 100, 100, date, time)>

Passbook<1, new Transactions(particulars, chequeNo, debitAmt, creditAmt, FinalBal, Date, time)>

So, as the user will perform transactions, no of entries in passbook will increase (transactions).

Every user will have its own copy of passbook object.

**\*** int sno :It will hold the serial no of transaction